# Practical Cryptanalysis of a Public-Key Encryption Scheme Based on New Multivariate Quadratic Assumptions

Martin R. Albrecht[1], Jean-Charles Faugere[2,3,4],

Robert Fitzpatrick[5], Ludovic Perret[2,3,4],

Yosuke Todo[6], Keita Xagawa[6]

1 Technical University of Denmark, 2 Sorbonne Universités, 3 INRIA, 4 CNRS,

5 Royal Holloway, University of London, 6 NTT Secure Platform Laboratories

# Summary

- We revisit an MQ-based cryptosystem proposed by Huang, Liu and Yang at PKC2012.
- We can regard HLY12 as lattice-based cryptosystems.
- A Core i7 PC finds the secret keys in 5 - 16 min by using LLL for proposed parameter sets.
- Recommendation parameters.

# Agenda

- Introduction
  - MQ-based cryptography
- The HLY12 Cryptosystem
- Attack for Lattice
- New Security Estimation
- Recommendation Parameters
- Conclusion

# MQ (Multivariate Quadratic Polynomials)

- Quantum computers break RSA, DH and so on.
- We are working on Post-Quantum cryptography
  - Code-based cryptography
  - Lattice-based cryptography
  - Multivariate-based cryptography

# MQ (Multivariate Quadratic Polynomials)

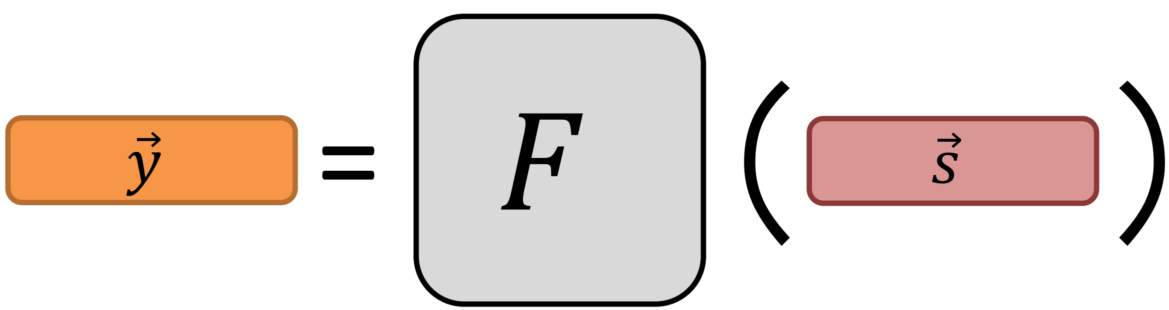
We let $\mathcal{Q} = \{f \in \mathbb{F}[x_1, \ldots, x_n] | \deg(f) \leq 2\}$

## MQ Problem

Input : $F = (f_1, \ldots, f_m) \in \mathcal{Q}^m$ and $\vec{y} = (y_1, \ldots, y_m) \in \mathbb{F}^m$

Output : $\vec{s} = (s_1, \ldots, s_n) \in \mathbb{F}^n$ s.t. $F(\vec{s}) = \vec{y}$

- This problem is NP-hard.



# Agenda

- Introduction
  - MQ-based cryptography
- The HLY12 Cryptosystem
- Attack for Lattice
- New Security Estimation
- Recommendation Parameters
- Conclusion

# Several MQ-based cipher's Idea

- Make F as a trapdoor function
  - Choose $G \in \mathcal{Q}^m$ which is easily invertible

$$F = \underbrace{T}_{\text{Affine maps}} \circ G \circ \underbrace{S}_{\text{Affine maps}}$$

$$\underbrace{\vec{y}}_{\text{Ciphertexts}} = \underbrace{F}_{\text{Public key}} \left( \underbrace{\vec{s}}_{\text{Plaintexts}} \right)$$

# Several MQ-based cipher's Idea

**Caution!** Several schemes are broken!
e.g. C*, SFLASH, MQQ, $\ell$-IC, …

$$F = T \circ G \circ S$$

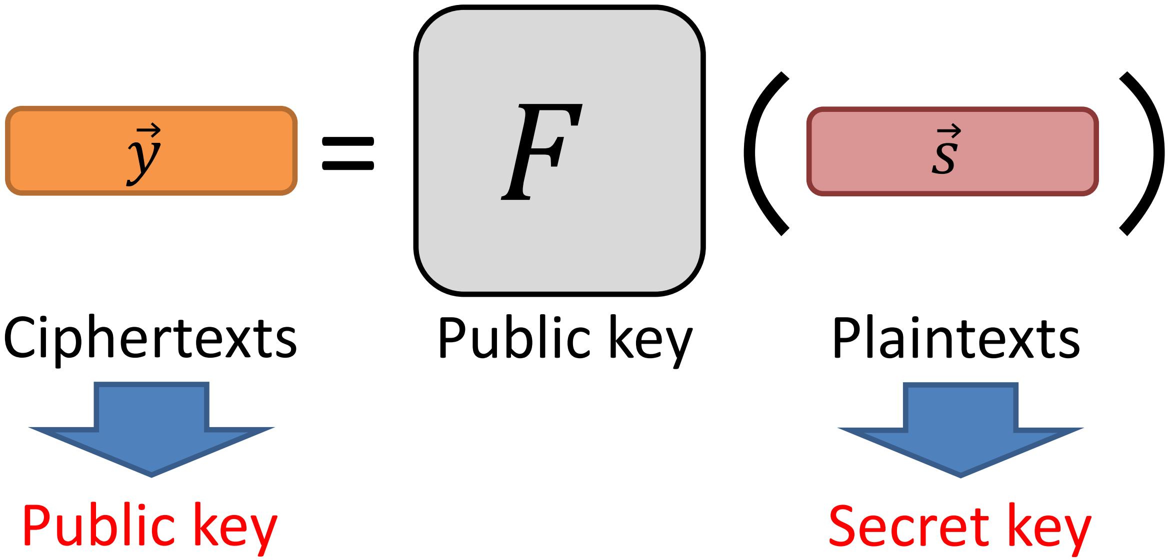
Affine maps ($T$) Affine maps ($S$)

$$\vec{y} = F(\vec{s})$$

Ciphertexts ($\vec{y}$) Public key ($F$) Plaintexts ($\vec{s}$)

# HLY12's Idea #1

- $F$ should be chosen randomly as possible
  - $F$ is NOT a trapdoor function.
  - Change the roles of $F, \vec{y}, \vec{s}$



# HLY12's Idea #2

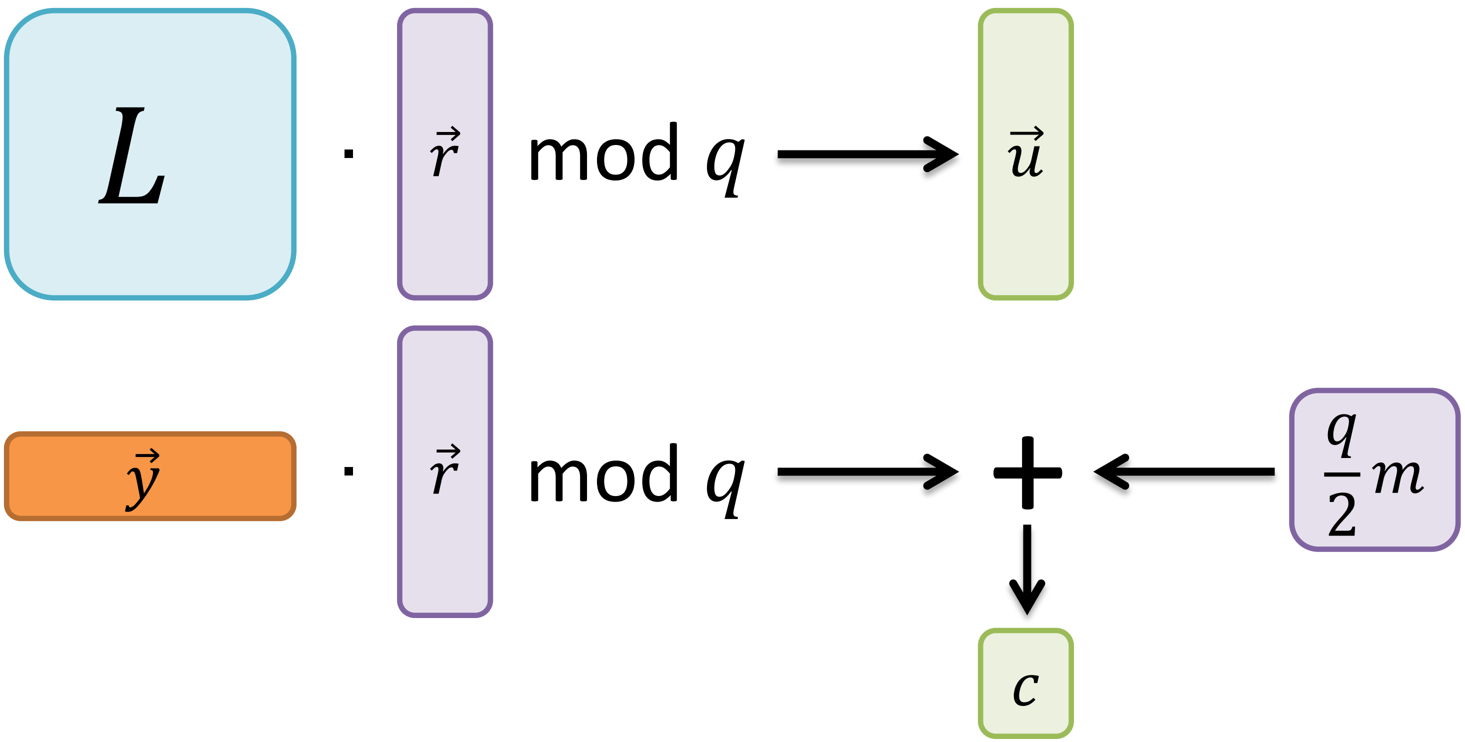
- $F$ consists of two parts

$$\vec{y} = F\left(\vec{s}\right)$$

$$= \vec{s} \cdot \underset{\text{Linear}}{L} + \underset{\text{Quadratic}}{Q(\vec{s})} \bmod q$$

# HLY12's Encryption

- Choose random $\vec{r}$



# HLY12's Decryption

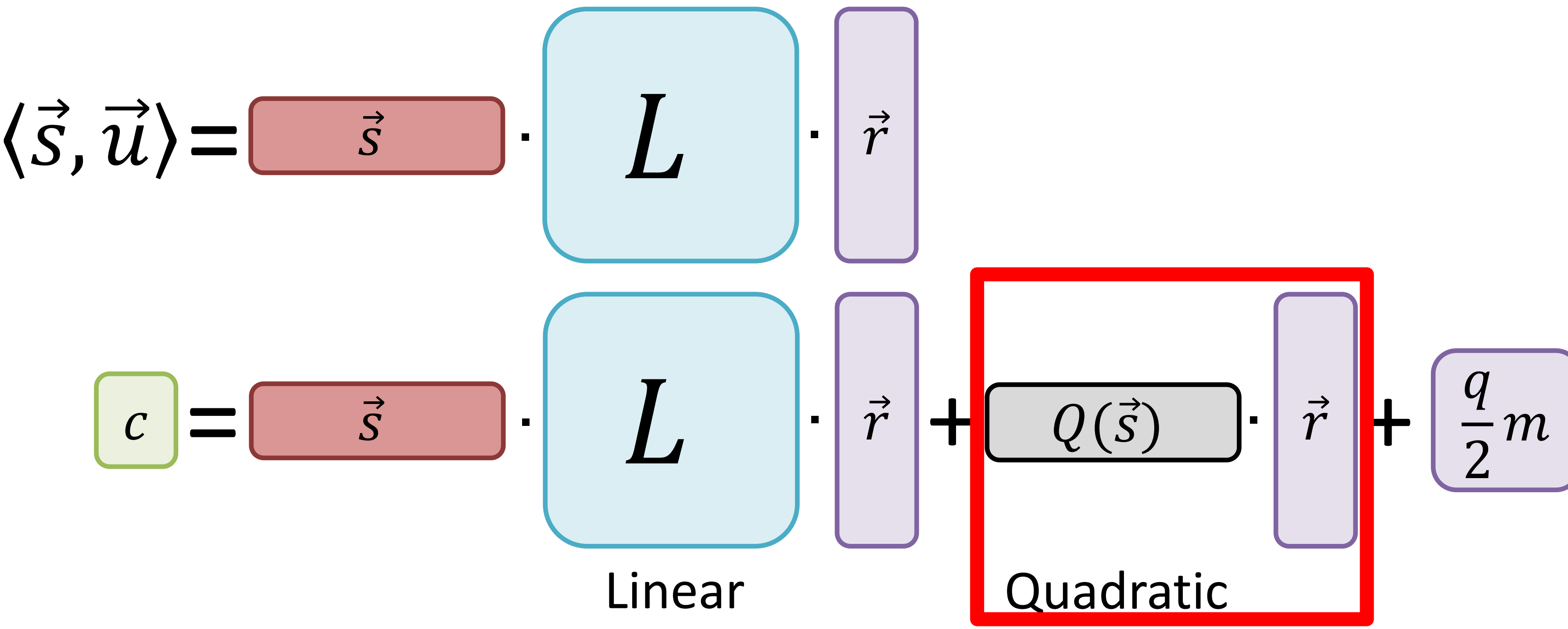
- The 1st term of $\langle \vec{s}, \vec{u} \rangle$ is the same as that of $c$

$$\langle \vec{s}, \vec{u} \rangle = \vec{s} \cdot L \cdot \vec{r}$$

$$c = \underbrace{\vec{s} \cdot L \cdot \vec{r}}_{\text{Linear}} + \underbrace{Q(\vec{s}) \cdot \vec{r}}_{\text{Quadratic}} + \frac{q}{2} m$$

# HLY12's Decryption

- If $Q(\vec{s}) \cdot \vec{r}$ is short, $m$ can be recovered.



# Suggested Parameters

| Case | n | m | q | Hardness $T\mu^{-1}$ |
|---|---|---|---|---|
| 1 | 200 | 400 | $\approx 2^{74}$ | $2^{256}$ $(2^{156}, 2^{-100})$ |
| 2 | 256 | 512 | $\approx 2^{76}$ | $2^{309}$ $(2^{205}, 2^{-104})$ |

$$\boxed{\vec{y}} = \boxed{\vec{s}} \cdot \boxed{L} + \boxed{Q(\vec{s})} \bmod q$$

$$\mathcal{U}([-2,2]) \qquad \mathcal{U}(\mathbb{Z}_q) \qquad (\vec{s}Q_1\overrightarrow{s^t}, \dots, \vec{s}Q_m\overrightarrow{s^t})$$

Given $(L, Q, \vec{y}) \in \mathbb{Z}_q^{n\times m} \times (\mathbb{Z}_q^{n\times n})^m \times \mathbb{Z}_q^m$, finding $\vec{s}$.

$(T, \mu)$ : no solver running in time less than $T$ can solve the system with prob. $\geq \mu$.

# Agenda

- Introduction
  - MQ-based cryptography
- The HLY12 Cryptosystem
- Attack for Lattice
- New Security Estimation
- Recommendation Parameters
- Conclusion

# Security of the HLY12

- The security is estimated by the XL algorithm.
  - Two recommendation parameters were given.
- We can regard HLY12 as lattice-based cryptosystems.
  - $Q(\vec{s})$ is very small vectors

# Lattice-based cryptography?

- We can regard $Q(\vec{s})$ as error vectors

$$c = \vec{s} \cdot \underbrace{L}_{\text{Linear}} \cdot \vec{r} + \boxed{\underbrace{Q(\vec{s}) \cdot \vec{r}}_{\text{Quadratic}}} + \frac{q}{2}m$$

$$\Downarrow$$

$$\vec{e}$$

**Observation** If we regard $Q(\vec{s})$ as error vectors, HLY12 is similar to the Regev Cryptosystem

# First Lattice (q-Ary Lattice)

$$\vec{y} = \vec{s} \cdot L + Q(\vec{s}) \bmod q$$

$$\begin{pmatrix} qI & \\ L & \\ \vec{y} & 1 \end{pmatrix} \Rightarrow \begin{pmatrix} Q(\vec{s}) & 1 \end{pmatrix}$$

| | |
|---|---|
| Case 1 | 26 hours |
| Case 2 | 3 days |

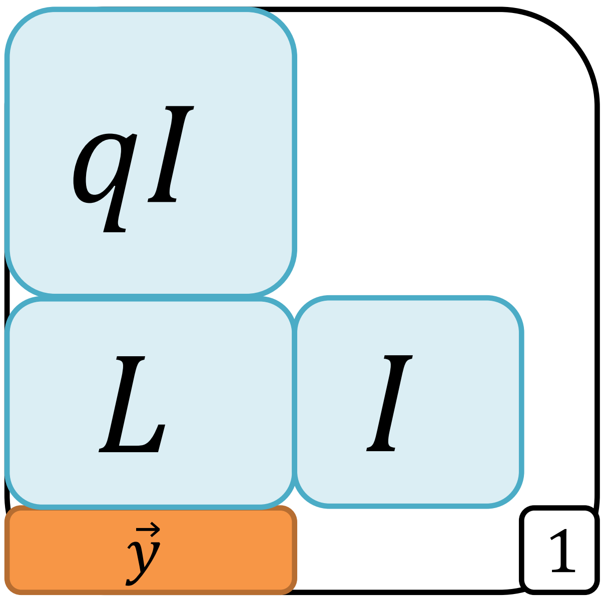
**Observation**

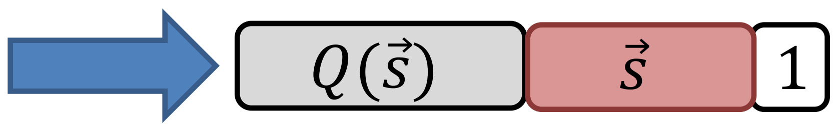
We can attack HLY12 in practical time by using lattice reduction algorithms

# Second Lattice (NTRU-like lattice)

- $\vec{s}$ is very short compared with $Q(\vec{s})$

$$Q(s)_i \in \lfloor N(0,10) \rceil$$
$$s_i \in \mathcal{U}([-2,2])$$



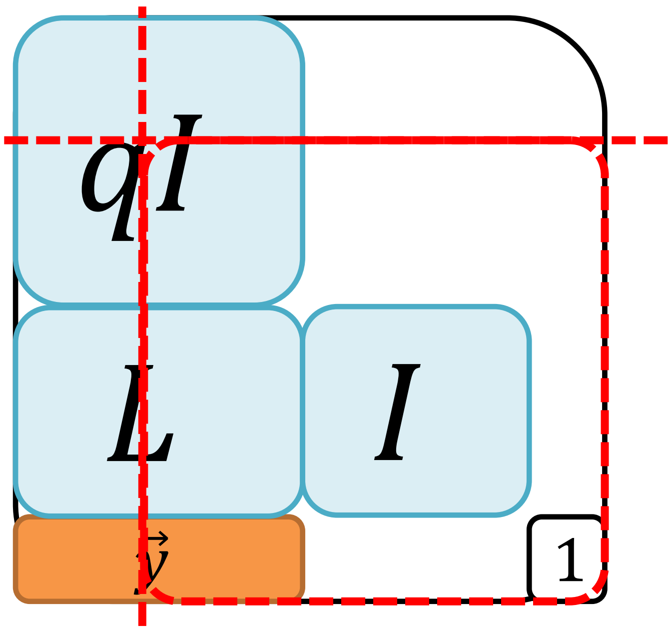



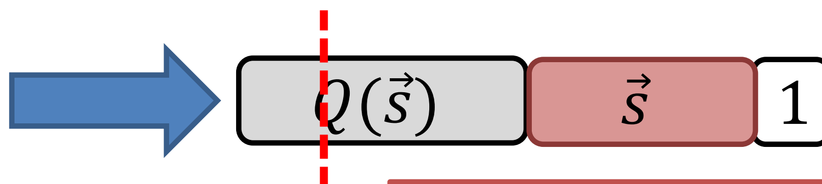

**Observation** The dimension is so huge…

# Third Lattice (Truncated lattice)

- We can truncate the matrix

$$Q(s)_i \in \lfloor N(0,10) \rceil$$
$$s_i \in \mathcal{U}([-2,2])$$




| | |
|---|---|
| Case 1 | 5 min |
| Case 2 | 16 min |

**Observation**

We should choose $s_i$ from $\lfloor N(0,10) \rceil$ to avoid our lattice attack.

# Agenda

- Introduction
  - MQ-based cryptography
- The HLY12 Cryptosystem
- Attack for Lattice
- New Security Estimation
- Recommendation Parameters
- Conclusion

Robert will talk the remaining contents

# Practical Cryptanalysis of a Public-Key Encryption Scheme Based on New Multivariate Quadratic Assumptions

Martin R. Albrecht [1] Jean-Charles Faugére [2,3]
Robert Fitzpatrick[4,5] Ludovic Perret[2,3]
Yosuke Todo[6] Keita Xagawa[6]

Technical University of Denmark

INRIA & CNRS
ISG, Royal Holloway, University of London,
IIS, Academia Sinica
NTT Secure Platform Laboratories

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

**How to estimate the practical security of LWE/LWE-like functions?**

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

**How to estimate the practical security of LWE/LWE-like functions?**

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

## How to estimate the practical security of LWE/LWE-like functions?

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

## How to estimate the practical security of LWE/LWE-like functions?

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

**How to estimate the practical security of LWE/LWE-like functions?**

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

**How to estimate the practical security of LWE/LWE-like functions?**

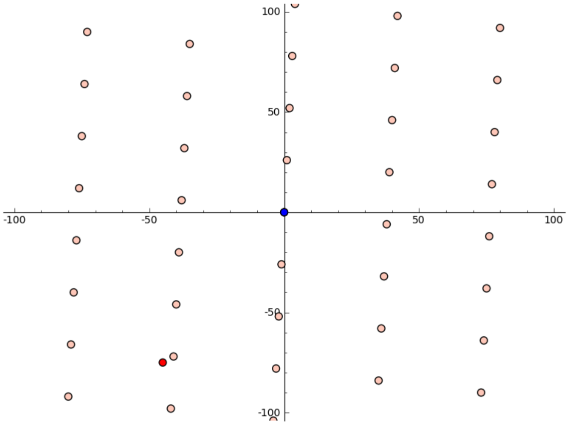
- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.

# Estimating LWE Security (i)

If we view HLY from an LWE perspective...

## How to estimate the practical security of LWE/LWE-like functions?

- In practise, by examining the cost of: dual-lattice-reduction + distinguishing (MR09); lattice-reduction + decoding (LP10, LN13) or embedding lattice reduction (AFG13).
- Dual-lattice distinguishing
- Reduction + decoding
- Embedding
- (and BKW)
- In general, security closely related to $q/\sigma$.


100
50
-50
-100
-100
-50
50
100

# Dual-Lattice Distinguishing

- Find a short $\vec{y} \in \mathcal{L}^{\perp}$ (scaled dual $q$-ary lattice): check if $\langle \vec{y}, \vec{c} \rangle = \langle \vec{y}, \mathbf{A}^T \vec{s} + \vec{e} \rangle = \langle \vec{y}, \vec{e} \rangle$ is short.
- Distinguishing advantage: $\varepsilon \approx \exp\left(-\pi \cdot (\|\vec{y}\| \cdot \sigma\sqrt{2\pi}/q)^2\right)$

# Dual-Lattice Distinguishing

- Find a short $\vec{y} \in \mathcal{L}^{\perp}$ (scaled dual $q$-ary lattice): check if $\langle \vec{y}, \vec{c} \rangle = \langle \vec{y}, \mathbf{A}^T \vec{s} + \vec{e} \rangle = \langle \vec{y}, \vec{e} \rangle$ is short.
- Distinguishing advantage: $\varepsilon \approx \exp\left(-\pi \cdot (\|\vec{y}\| \cdot \sigma\sqrt{2\pi}/q)^2\right)$

# Reduction + Decoding

- Reduce the primal basis
- Then carry out Klein's algorithm to find closest vector (or a pruned version [LN13])
- Most effective method in practice

# Reduction + Decoding

- Reduce the primal basis
- Then carry out Klein's algorithm to find closest vector (or a pruned version [LN13])
- Most effective method in practice

# Reduction + Decoding

- Reduce the primal basis
- Then carry out Klein's algorithm to find closest vector (or a pruned version [LN13])
- Most effective method in practice

# Embedding and BKW

- Embedding attack: Given a matrix-LWE sample $(\mathbf{A}, \vec{c})$ we construct
$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \overline{\mathbf{A}} \\ \mathbf{0} & q\mathbf{I} \end{pmatrix} \mathbf{P}^{-1}$$
Then construct
$$\mathbf{B} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{t} & t \end{pmatrix}$$
- $[\mathbf{t} \quad t]$ shortest vector in $\mathcal{L}(\mathbf{B})$. Second minimum is first minimum of $\mathcal{L}(\mathbf{A}')$. Resulting unique-SVP instance somehow easier...
- BKW: previous talk - also breaks the proposed parameters but not as effectively as lattice attacks

# Embedding and BKW

- Embedding attack: Given a matrix-LWE sample $(\mathbf{A}, \vec{c})$ we construct

$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \overline{\mathbf{A}} \\ \mathbf{0} & q\mathbf{I} \end{pmatrix} \mathbf{P}^{-1}$$

Then construct

$$\mathbf{B} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{t} & t \end{pmatrix}$$

- $[\mathbf{t} \quad t]$ shortest vector in $\mathcal{L}(\mathbf{B})$. Second minimum is first minimum of $\mathcal{L}(\mathbf{A}')$. Resulting unique-SVP instance somehow easier...
- BKW: previous talk - also breaks the proposed parameters but not as effectively as lattice attacks

# Embedding and BKW

- Embedding attack: Given a matrix-LWE sample $(\mathbf{A}, \vec{c})$ we construct

$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \overline{\mathbf{A}} \\ \mathbf{0} & q\mathbf{I} \end{pmatrix} \mathbf{P}^{-1}$$

  Then construct

$$\mathbf{B} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{t} & t \end{pmatrix}$$

- $[\mathbf{t} \quad t]$ shortest vector in $\mathcal{L}(\mathbf{B})$. Second minimum is first minimum of $\mathcal{L}(\mathbf{A}')$. Resulting unique-SVP instance somehow easier...
- BKW: previous talk - also breaks the proposed parameters but not as effectively as lattice attacks

# Embedding and BKW

- Embedding attack: Given a matrix-LWE sample $(\mathbf{A}, \vec{c})$ we construct
$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \overline{\mathbf{A}} \\ \mathbf{0} & q\mathbf{I} \end{pmatrix} \mathbf{P}^{-1}$$
Then construct
$$\mathbf{B} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{t} & t \end{pmatrix}$$
- $[\mathbf{t} \quad t]$ shortest vector in $\mathcal{L}(\mathbf{B})$. Second minimum is first minimum of $\mathcal{L}(\mathbf{A}')$. Resulting unique-SVP instance somehow easier...
- BKW: previous talk - also breaks the proposed parameters but not as effectively as lattice attacks

# Embedding and BKW

- Embedding attack: Given a matrix-LWE sample $(\mathbf{A}, \vec{c})$ we construct
$$\mathbf{A}' = \begin{pmatrix} \mathbf{I} & \overline{\mathbf{A}} \\ \mathbf{0} & q\mathbf{I} \end{pmatrix} \mathbf{P}^{-1}$$
Then construct
$$\mathbf{B} = \begin{pmatrix} \mathbf{A}' & \mathbf{0} \\ \mathbf{t} & t \end{pmatrix}$$
- $[\mathbf{t} \quad t]$ shortest vector in $\mathcal{L}(\mathbf{B})$. Second minimum is first minimum of $\mathcal{L}(\mathbf{A}')$. Resulting unique-SVP instance somehow easier...
- BKW: previous talk - also breaks the proposed parameters but not as effectively as lattice attacks

# Estimating LWE Security (ii)

Simply, characterise ‘strength’ of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ-20}} \approx 1.0128$

$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: “well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110$

# Estimating LWE Security (ii)

Simply, characterise ‘strength’ of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ-20}} \approx 1.0128$
$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: “well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110$

# Estimating LWE Security (ii)

Simply, characterise ‘strength’ of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ-20}} \approx 1.0128$
$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: “well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110$

# Estimating LWE Security (ii)

Simply, characterise 'strength' of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ}-20} \approx 1.0128$
$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: "well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8 / \log_2 \delta_0 - 110$

# Estimating LWE Security (ii)

Simply, characterise 'strength' of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ-20}} \approx 1.0128$
$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: "well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110$

# Estimating LWE Security (ii)

Simply, characterise ‘strength’ of lattice reduction by Hermite root factor, $\delta_0$. $\delta_0^{\mathrm{LLL}} \approx 1.0219$, $\delta_0^{\mathrm{BKZ-20}} \approx 1.0128$
$\delta_0 = 1.009$: roughly limit of current algorithms. $\delta_0 = 1.005$: “well-beyond reach".

## Running time of BKZ?

- Still problematic to predict - too many variables. Block-size, choice of SVP sub-routine (further variables), pre-processing of local bases, early termination etc.
- BKZ 2.0 simulator, simple model of Lindner & Peikert
- $\log_2 T_{\mathrm{sec}} = 1.8/\log_2 \delta_0 - 110$

# HLY Security Conditions (i)

## HLY Conditions

- $k \cdot \zeta \cdot n^{2+\lambda} \cdot m \cdot \beta^2 \leq q/4$ (correct decryption)
- $m \cdot \log(2n^\lambda + 1) \geq (n+1)\log q + 2k$ (hardness of subset sum problem)
- $n, m, q, \zeta, \beta$ satisfy MQ hardness assumption

For security against the distinguishing attack:

## LWE-derived Conditions

- $\exp\left(-\frac{\pi^2}{12\beta^2} \cdot (ck)^{-2} \cdot n^{-4} \cdot 2^{3.6cn/(\tau+78.9)}\right) = d$

# HLY Security Conditions (i)

## HLY Conditions

- $k \cdot \zeta \cdot n^{2+\lambda} \cdot m \cdot \beta^2 \leq q/4$ (correct decryption)
- $m \cdot \log(2n^\lambda + 1) \geq (n+1) \log q + 2k$ (hardness of subset sum problem)
- $n, m, q, \zeta, \beta$ satisfy MQ hardness assumption

For security against the distinguishing attack:

## LWE-derived Conditions

- $\exp\left(-\frac{\pi^2}{12\beta^2} \cdot (ck)^{-2} \cdot n^{-4} \cdot 2^{3.6cn/(\tau+78.9)}\right) = d$

# HLY Security Conditions (i)

## HLY Conditions

- $k \cdot \zeta \cdot n^{2+\lambda} \cdot m \cdot \beta^2 \leq q/4$ (correct decryption)
- $m \cdot \log(2n^\lambda + 1) \geq (n+1)\log q + 2k$ (hardness of subset sum problem)
- $n, m, q, \zeta, \beta$ satisfy MQ hardness assumption

For security against the distinguishing attack:

## LWE-derived Conditions

- $\exp\left(-\frac{\pi^2}{12\beta^2} \cdot (ck)^{-2} \cdot n^{-4} \cdot 2^{3.6cn/(\tau+78.9)}\right) = d$

# Implications for Required Key Sizes

To reconcile HLY with security against the distinguishing attack, we have the following:

- 80-bit security $\Rightarrow$ $(n = 1140)$ $\Rightarrow$ public-key size: 1.03 GB
- 128-bit security $\Rightarrow$ $(n = 1530)$ $\Rightarrow$ public-key size: 2.49 GB

## Conclusions

- Scheme of HLY represents an interesting and rigorous approach to construct a provably-secure MQ PKC.
- Commendable that concrete parameters were proposed.
- However the extra structure required to describe it as MQ instead of LWE leads to prohibitive key sizes
- Ring-LWE analogue?

Questions?